

## Objectifs

- Savoir les caractéristiques du [microcontrôleur PIC16F887](#)
- Savoir lire le convertisseur A/N (ADC)
- Savoir comment afficher la valeur du convertisseur ADC en utilisant les LEDs

## Applications

- Instruments numériques (voltmètre, ampèremètre, wattmètre, etc.)
- Mesure de la [résistance](#)
- Surveillance d'un équipement [électronique](#) (surcharge, surintensité, etc.)
- Etc.



# Caractéristiques du microcontrôleur PIC16F887

- 35 instructions
- Horloge : DC au 20MHz
- Mode faible consommation
- POR : Powe on Reset
- PWRT : Power up Timer
- Mémoire [SRAM](#): 368
- Mémoire EEPROM 256
- Entrées/sorties : 35
- Convertisseur A/N : 14 canaux, 10 bits
- EUSART : 1
- MSSP : 1
- Compérateurs : 2
- Timers 8/16 bits : 2+1
- Standby Current: - 50 nA @ 2.0V, typical
- Operating Current: - 11 µA @ 32 kHz, 2.0V, typical - 220 µA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current: - 1 µA @ 2.0V, typical
- Autres: [Datasheet PIC16F887](#)

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	ECCP/ CCP	EUSART	MSSP	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F882	2048	128	128	28	11	1/1	1	1	2	2/1
PIC16F883	4096	256	256	24	11	1/1	1	1	2	2/1
PIC16F884	4096	256	256	35	14	1/1	1	1	2	2/1
PIC16F886	8192	368	256	24	11	1/1	1	1	2	2/1
PIC16F887	8192	368	256	35	14	1/1	1	1	2	2/1



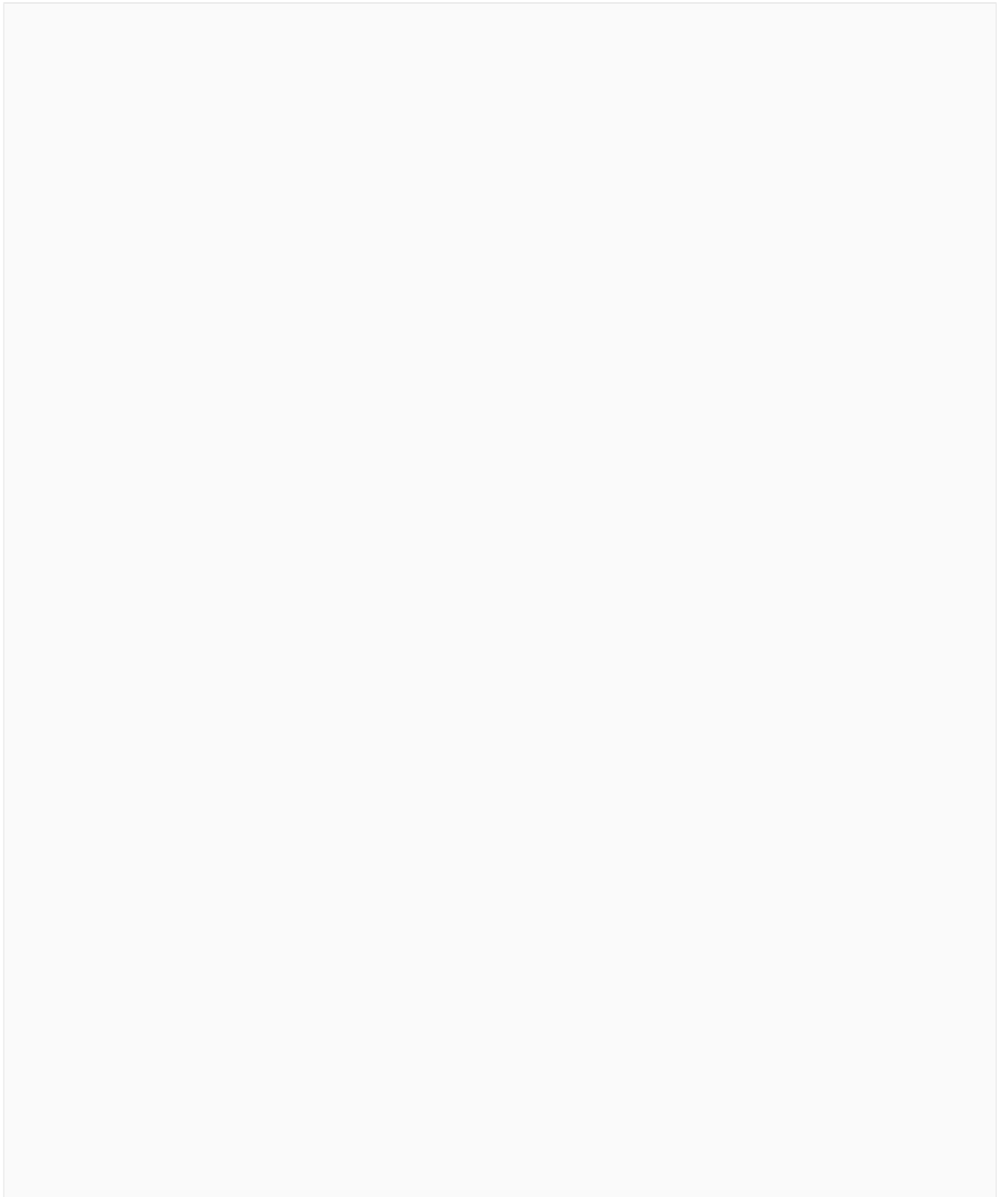
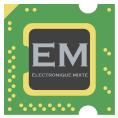
```
void ADC_Init();
```

Fonction d'initialisation du convertisseur A/N interne du PIC afin de fonctionner avec l'horloge RC. La fonction ne retourne aucune valeur et ne prend aucun argument à l'entrée. Elle nécessite la présence d'un convertisseur A/N dans l'architecture matérielle du PIC.

```
unsigned ADC_Get_Sample(unsigned short  
channel);
```

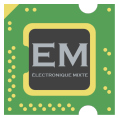
La fonction prend en entrée le numéro du canal, puis elle retourne la valeur actuelle du convertisseur en format unsigned short (16 bits non signés ou deux octets). En effet, les convertisseurs A/N des PICs ont souvent une résolution sur 8,10 ou 12 bits (inférieure à 16 bits). Avant d'utiliser la fonction il faut faire appel à la fonction d'initialisation: ADC\_Init() et s'assurer que le pin est bien configuré en entrée (TRISx=1 ).

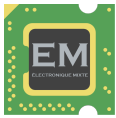
**Exemple:**





```
unsigned adc_val;  
...  
adc_val = ADC_Get_Sample(2);    // read le canal 2 du convertisseur
```





## unsigned ADC\_Read(unsigned short channel);

Identique à la fonction précédente. En revanche. L'appel de la fonction ADC\_Init n'est pas obligatoire.

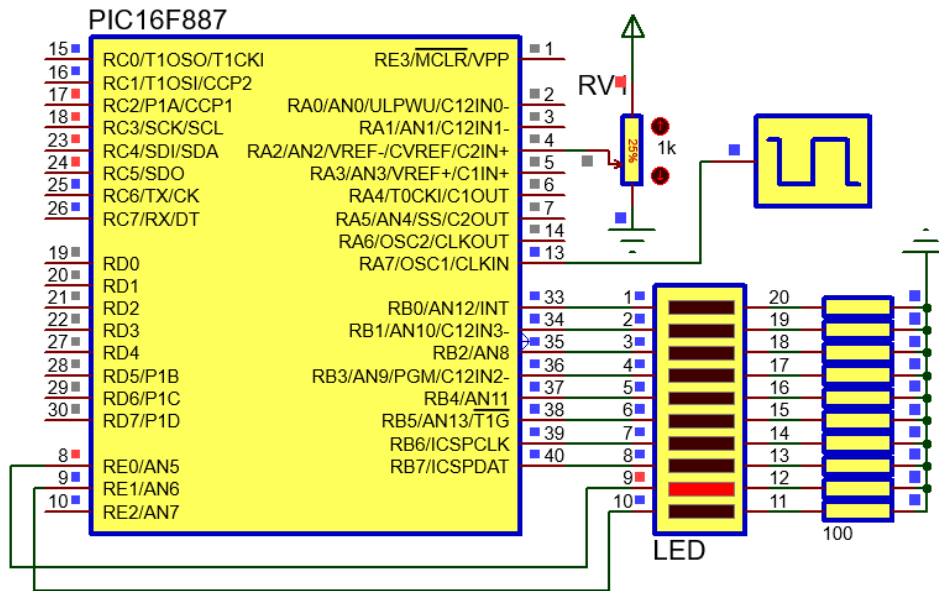
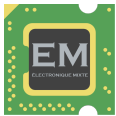
## Fonctionnement du Clignotant à LEDs

Le fonctionnement de l'application consiste le contrôle de l'allumage d'une série de 10 LEDs en utilisant un [potentiomètre](#) analogique. Le potentiomètre est branché avec le pin 2 du port A (PA2) configuré en entrée analogique. La valeur acquise par le convertisseur codée sur 10 bits (1024 combinaisons) sera ensuite affichée dans les LEDs. L'affichage sera effectué en deux étapes : Affichage du poids fort (MSB) de la valeur dans le port B, le deux bits restants (MSB) sera affiché dans le port C.

Nous avons utiliser une boucle infinie [while\(\)](#) afin de lire et afficher la valeur analogique dans les ports B et C branchés avec les 10 LEDs. Ci-dessous le programme [MikroC](#) et le schéma [ISIS](#) de câblage. L'allumage des [LED](#) est directement lié avec le pourcentage du curseur du potentiomètre RV1.

## Affichage en fonction du potentiomètre

- **25%**



• 50%

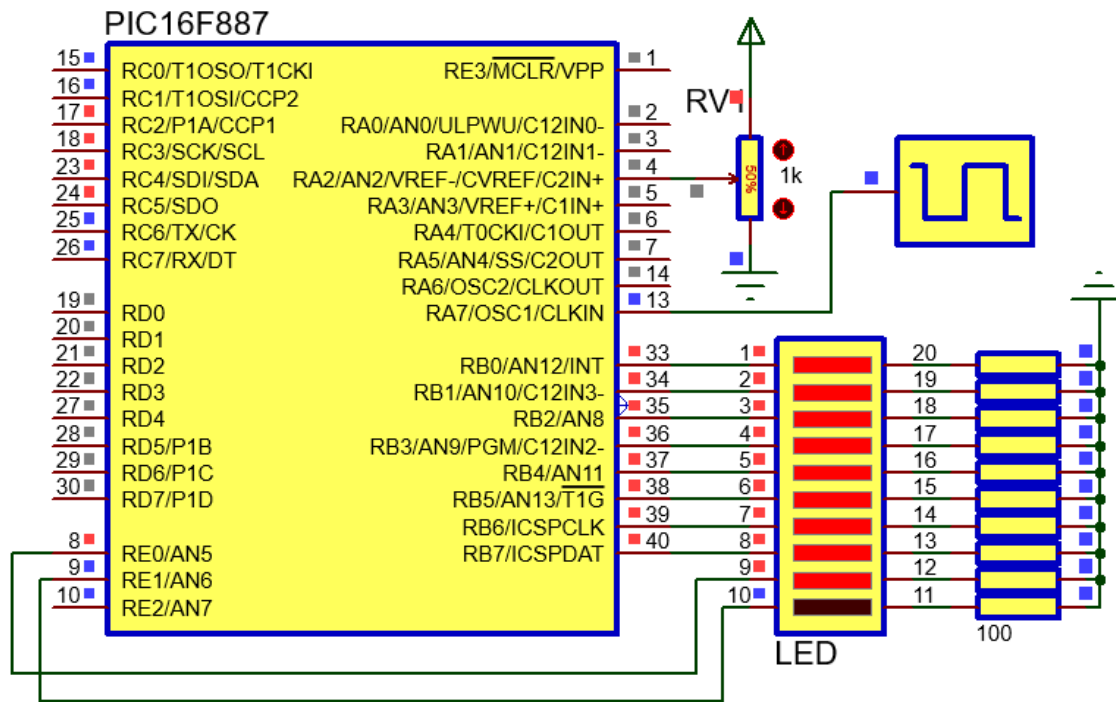
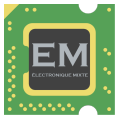




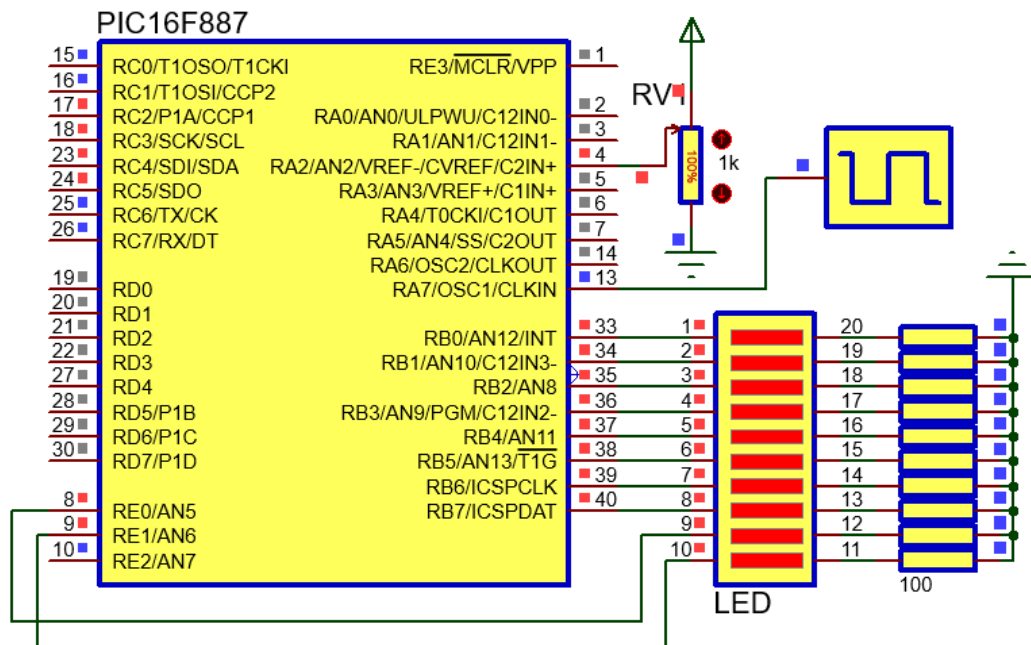
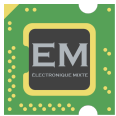
Clignotant à LED PIC16F887 et ADC



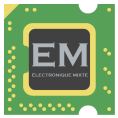
Clignotant à LED PIC16F887 et ADC

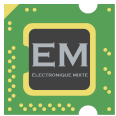


• 100%



## Programme MikroC





```
#include <built_in.h>

unsigned int adc_val;

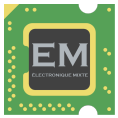
void main() {

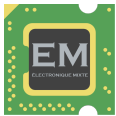
    // Initialisation ADC
    ADC_Init();

    // Port A en entrée
    TRISA=0xFF;

    // Configuration en sortie des ports
    TRISB=0;
    TRISC=0;
    TRISE=0;

    // Lecture et affichage de la donnée ADC
    while(1)
    {
        adc_val=ADC_Read(2); // Ou adc_val=ADC_Get_Sample(2);
        PORTB=adc_val;       // Affichage 8 bits LSB
        PORTE=adc_val>>8;   // Affichage 8 bits MSB
    }
}
```





Clignotant à LED PIC16F887 et ADC

# Téléchargement

- [Programme MikroC](#)
- [Schéma ISIS Proteus](#)

[Retour à l'accueil MikroC](#)